

LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A239 253



DTIC ACCESSION NUMBER

LEVEL

INVENTORY

WL-TR-91-8024

DOCUMENT IDENTIFICATION

Jul 91

DISTRIBUTION STATEMENT

ACCESSION FOR

NTIS GRA&I

DTIC TRAC

UNANNOUNCED

JUSTIFICATION

BY

DISTRIBUTION/

AVAILABILITY CODES

DISTRIBUTION

AVAILABILITY AND/OR SPECIAL

A-1

DISTRIBUTION STAMP



DATE ACCESSIONED

DATE RETURNED

91 8 98 006

DATE RECEIVED IN DTIC

91-07299



REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

H
A
N
D
L
E

W
I
T
H

C
A
R
E

AD-A239 253



PRODUCT DEFINITION DATA INTERFACE (PDDI)

Operator's Manual

Jeffrey Altemueller
John Purses
George White
Rick Helldoerfer

McDonnell Aircraft Company
McDonnell Douglas Corporation
P. O. Box 516
St. Louis, MO 63166

July 1991

Final Report

Approved for public release; distribution is unlimited.

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY UNCLASSIFIED			3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5. MONITORING ORGANIZATION REPORT NUMBER(S) WL-TR-91-8024		
6a. NAME OF PERFORMING ORGANIZATION McDonnell Aircraft Company		6b. OFFICE SYMBOL (if applicable) MCAIR		7a. NAME OF MONITORING ORGANIZATION Manufacturing Technology Dir. (WL/MTIB) Wright Laboratory	
6c. ADDRESS (City, State, and ZIP Code) McDonnell Douglas Corporation P. O. Box 516, St. Louis, MO 63166			7b. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433-6533		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (if applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-82-C-5036	
8c. ADDRESS (City, State, and ZIP Code) Wright Patterson Air Force Base, Ohio 45433-6533			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. 78011F	PROJECT NO. 3095	TASK NO. 06
			WORK UNIT ACCESSION NO. 29		
11. TITLE (Include Security Classification) PRODUCT DEFINITION DATA INTERFACE (PDDI) Operator's Manual					
12. PERSONAL AUTHOR(S) Jeffrey Altemueller, Purses John, White George, Helldoerfer Rick					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM Dec 86 to Sep 87		14. DATE OF REPORT (Year, Month, Day) July 1991	
				15. PAGE COUNT 55	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
1308	0905		Product Definition Data Life Cycle Document Engrg./Mfg. Interface		
			ICAM Architecture CAD/CAM (continued on back)		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This document is the Operator's Manual for the Product Definition Data Interface (PDDI) Extensions. This document provides procedures for installation of the PDDI Software.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Alan Winn			22b. TELEPHONE (Include Area Code) (513) 255-8787		22c. OFFICE SYMBOL WL/MTIB

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.




ALAN R. WINN
Project Manager

30 May '91

DATE

FOR THE COMMANDER:



BRUCE A. RASMUSSEN, Chief
Integration Technology Division
Manufacturing Technology Directorate

31 May 91

DATE

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/MTIB, WPAFB, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

FOREWORD

This document was produced under Air Force Contract F33615-82-C-5036, Product Definition Data Interface (PDDI). This contract is sponsored by the Air Force Wright Aeronautical Laboratories, Materials Laboratory, Air Force Systems Command, Wright-Patterson, Air Force Base, Ohio.

This program is being administered under the technical direction of Lt. Eric Gunther, ICAM Project Manager. The MCAIR Program Manager is Mr. Jerry Weiss and Mr. Herb Ryan is the Deputy Program Manager.

This document was prepared in accordance with the ICAM Configuration Management Life Cycle Documentation requirements for the Configuration Item.

OPERATOR'S MANUAL

TABLE OF CONTENTS

		<u>PAGE</u>
Section 1	SCOPE.	1-1
1.1	Identification	1-1
1.2	Introduction	1-1
1.3	Approach	1-2
Section 2	REFERENCES	2-1
2.1	Applicable Documents	2-1
2.1.1	Specifications	2-1
2.1.2	Standards.	2-1
2.1.3	Other Publications	2-1
2.2	Terms and Abbreviations.	2-3
Section 3	SYSTEMS OPERATIONS	3-1
3.1	System Overview	3-1
3.1.1	system Interfaces.	3-5
3.2	System Environment	3-7
3.2.1	Schema Manager	3-7
Section 4	INSTALLATION	4-1
4.1	Overview	4-1
4.2	Operations	4-1
4.2.1	Introduction	4-1
4.2.2	Data Sets.	4-2
4.2.3	Operator Interface	4-2
4.2.4	Compile.	4-2
4.2.5	Link	4-2
4.3	Installation Procedures.	4-2
4.3.1	Introduction	4-2
4.3.2	Preliminary Steps.	4-3
4.3.3	Procedures - Translator.	4-3
4.3.4	Procedures - Access Software Only.	4-4
Appendix A	CLISTS and JCL	A-1
Appendix B	IBM to VAX Conversion Procedures	B-1
Appendix C	Data Stores.	C-1
Appendix D	Messages	D-1

LIST OF ILLUSTRATIONS

Figure 3-1	PDDI System Architecture	3-2
Figure 3-2	Translator Architecture.	3-5
Figure 3-3	Native Database Access	3-6
Figure 3-4	PDDI Environment	3-8
Figure 3-5	Schema Manager Architecture.	3-9

SECTION 1

SCOPE

1.1 Identification

This Operator's Manual describes the system operating commands and software installation procedures for the Product Definition Data Interface (PDDI) Project 5601. This project was developed under Air Force Contract F33516-82-C-5036.

1.2 Introduction

This manual describes the system operation commands and installation procedures. It is intended for use by computer operators and programming personnel. It is assumed that personnel installing this software are familiar with the "native" system installation requirements and procedures.

The associated PDDI User's Manuals (UM) provide a guide for use of the PDDI System Translator, PDDI Access Software, and Schema Manager Software.

Capabilities documented in the Translator UM include "PUT" a PDDI model into the Working Form and "GET" a PDDI model from the Working Form via the Access Software using the PDDI translator.

An Access Software UM provides a guide for application programmers to use the Access Software. Capabilities described in this document are Access Software Initialization, Entity Creation, Deletion and Manipulation, and List Operations.

The Schema Manager UM describes how to define CAD/CAM entities suitable for the Working Form of the PDDI Access Software.

The PDDI Product Specification provides routine descriptions, Data Dictionary Listings and PDDI System messages for system maintenance purposes.

The PDDI software was designed to be transportable and has been operated on IBM 43xx, 308x or DEC VAX 11/780 computers. These environmental requirements are described in Section 3 of this document. The PDDI system document does not address local (native) system or computing environment documentation.

This manual will address IBM procedures and terminology only. Appendix C provides the IBM to VAX conversion procedures used during development of this PDDI prototype system.

1.3 Approach

This Operator's Manual is divided into four (4) main sections: Scope, References, System Operations, and System Installation. Appendices provide supplemental information.

Section 1 - Scope of this document.

Section 2 - Reference documentation applicable to PDDI and this document.

Section 3 - The PDDI architecture at a high level, system environmental needs, and system interfaces.

Section 4 - System installation procedures to install PDDI on IBM 43xx computers and the running of the TSO/MVS operating system.

Appendix A - Listing of CLISTS and JCL used for system installation.

Appendix B - Procedures for IBM to VAX conversion.

Appendix C - Datasets used and a description of each.

Appendix D- Listing of PDDI system messages.

SECTION 2

REFERENCES

2.1 Applicable Documents

2.1.1 Specification:

DOD-D-1000B	Drawings, Engineering and Associated Lists
MIL-D-5840	Requirements for Data, Engineering and Technical Reproduction

2.1.2 Standards:

ANSI Y14.5	Dimensioning and Tolerancing
ANSI Y14.26M	Digital Representation Communication of Production Definition Data
ANSI B46.1	Surface Texture (Surface Roughness, Waviness and Lay)
ANSI B92.1	Involute Splines and Inspection
DOD-STD-100C	Engineering Drawing Practices
MIL-STD-9	Screw Thread Conventions and Methods of Specifying
MIL-STD-12	Abbreviations for Use on Drawings, Specifications, Standards and in Technical Documents
IDS150120000C	ICAM Documentation Standards
IEEE STD 829	Standards for Software Test Documentation
ISO TC184/SC4/WG1	The STEP File Structure (Working Paper) 28 April 1986

2.1.3 Other Publications:

CLD150120000	ICAM Document Catalog
FTR110210000U	ICAM Architecture
FTR110232000U	ICAM Architecture Part II, Automated IDEFO Development

Product Definition Data Interface

ITR560130001U	First Interim Technical Report (Period 1 Oct 82 - 31 Dec 82)
ITR560130002U	Second Interim Technical Report (Period 1 Jan 83 - 31 Mar 83)
ITR560130003U	Third Interim Technical Report (Period 1 Apr 83 - 30 June 83)
ITR560130004U	Fourth Interim Technical Report (Period 1 Jul 83 - 30 Sep 83)
ITR560130005U	Fifth Interim Technical Report (Period 1 Oct 83 - 1 Dec 83)
ITR560230006U	Sixth Interim Technical Report (Period 1 Jan 84 - 31 Mar 84)
ITR560130007U	Seventh Interim Technical Report (Period 1 Apr 84 - 30 Jun 84)
ITR560130008U	Eighth Interim Technical Report (Period 1 Jul 84 - 30 Sep 84)
ITR560130009U	Ninth Interim Technical Report (Period 1 Oct 84 - 31 Dec 84)
ITR560130010U	Tenth Interim Technical Report (Period 1 Jan 85 - 31 Mar 85)
FTR560130001U	Final Report - System Test Methodology, Volume III Technical Operating Report - Product Assurance/Quality Assurance - 15 Oct 84
SD 560130001U	Scoping Document
NAD560130000	Needs Analysis Document
SAD560130000	State-of-the-Art Document
SRD560130000	System Requirement Document

SDS560130000	System Design Specification Document
SS 560130100	System Specification Document
SS 560130200	System Specification Document - Draft Standard
PS 560130000A	Product Specification
UM 560130000B	User's Manual - Translator
UM 560130001	User's Manual - Access Software
STP560130000	System Test Plan
STR560130000	System Test Report

2.2 Terms and Abbreviations

ACCESS SOFTWARE - A set of routines for creating, managing and querying an incore Working Form model.

ANSI - American National Standards Institute.

APPLICATION - Refers generically to any software modules which are used in CAD/CAM functions.

APPLICATION REQUEST - A request initiated by an application program, either through batch or interactive processing, which will interrogate the model through the PDDI Access Software to obtain or operate on specific information regarding the model and its components or elements.

APPLICATION REQUESTED DATA - The data which fulfills the application's original request and which is in the proper format and readable by the application.

ASCII - American Standard Code for Information Interchange.

ATTRIBUTE - An item of information about an entity. A key attribute identifies the entity; a role attribute gives a fact about an entity.

CAD/CAM - Computer Aided Design/Computer Aided Manufacturing.

CLASS - A collection of entities that are alike in some manner.

CLIST - IBM Command lists.

CONSTITUENT - A specific instance of an entity that is used in the definition of some other entity.

DOMAIN - The set of values permissible in a given context. A natural domain is the value set native to a given machine architecture; an imposed domain is a specific subset of the natural domain.

DYNAMIC ALLOCATION - The allocation (and deallocation) of memory resources as required by the application. The opposite is static allocation where a fixed size segment of memory is available to the application.

EBCDIC - Extended Binary Coded Decimal Interchange Code (IBM character set).

ENTITY - A collection of facts (attributes) about something of interest.

EXTERNAL REFERENCE - A reference to some quantity of data that exists somewhere outside the scope of the immediate body of information.

FUNCTIONALITY - (1) To show that the configuration item has fulfilled the specified requirements. (2) The receiving and sending systems can operate on the entity in the same manner with the same results within a pre-defined tolerance.

INCLUDE FILE - Pascal source code from another file or library included on the compilation of a Pascal source file.

INPUT DATA - That information which the application needs to supply in order to interrogate or operate on the model. This data may assume only these forms prescribed by the PDDI Access Software specifications.

INTERPRETED REQUEST - Input data which has been appropriately modified to conform to the PDDI Access Software's internal data representation so that it may be further processed.

JCL - Job Control Language - IBM language used to identify a job and describe its requirements to an operating system.

KEY - An item of data that uniquely identifies some specific instance of an entity.

MAS - MCAIR's acronym for the PDDI Access Software (Model Access System).

METAMODEL - A body of data that defines the characteristics of a data model or structure.

MODEL - A collection of PDD that is transferable, displayable, accessible, and equivalent to a Part. The internal representation of the application data, as initiated and organized by the user. The model is also referred to as the Working Form.

MODEL NETWORK DEFINITION - The set of rules and definitions which outline in detail the data structure whereby higher order entities may be composed of lower order entities, or constituents, and the lower order entities may be constituents of one or more higher order entities.

NATIVE SYSTEM - The PDD and applications in a format that is unique to the database of a CAD system.

PARSE - The process of analyzing input strings (records) to identify fields and to verify that the data is in a valid format.

PDD - Product Definition Data.

POST-PROCESSOR - A phase of the translator where data is received from the Exchange Format and is converted to the Working Form.

PRE-PROCESSOR - A phase of the translator where data is taken from the Working Form and is converted to the Exchange Format.

QUALITY - The composite of all the attributes or characteristics including performance of an item or product.

QUALITY ASSURANCE - The planned and systematic establishment of all actions (management/engineering) necessary to provide adequate confidence and nonconformance prevention provisions. Reviews are established during the design phase and performed throughout the software development and life cycle phases.

QUALITY CONTROL - The planned and systematic application of all actions (management/technical) necessary to control raw materials or products through the use of test, inspect, evaluate, and control of processes.

REQUESTED DATA - See Application Requested Data.

RUN SYSTEM - The Translator sub-package which provides the communication interface between the user and the pre/post-processors.

SCHEMA - Those definitions which describe the content of the data and the relationship between the various elements or components of the data.

SCHEMA MANAGER (SCE) - The software for creating, managing, and querying entity definitions.

SOFTWARE QUALITY ASSURANCE (SQA) - The planned and systematic establishment of all actions necessary to provide adequate confidence that nonconformance prevention provisions and reviews are established during the design phase and performed throughout the software development and life cycle phases.

SOFTWARE QUALITY ASSURANCE PLAN (SOAP) - An organized description of the methods, policies, and procedures necessary to conduct software quality assurance and control activities during the design, development, delivery, and maintenance phases.

SOFTWARE QUALITY CONTROL - The planned and systematic application of all actions (management/technical) necessary to ensure that the software under development or maintenance satisfies the technical requirements through the use of tests, demonstrations, inspections, evaluations, and control of processes.

SYSTEM CONSTRAINTS - Those hardware and software environmental constraints which will be imposed upon PDDI Software that will limit its implementation and application. An example of such constraints might be the particular compiler used to compile the PDDI Access Software package.

TRANSFER DATA - The data required to make an exchange of data between systems (e.g., delimiters, record counts, record length, entity counts, numeric precision).

TRANSLATOR - A software package that is used for passing data between the Exchange Format and Working Form.

TSO - Time Sharing Option - IBM function which provides conversational time sharing from remote terminals.

USER COMPUTER SYSTEM - The specific hardware, operating systems, and applications software systems that the user will employ to implement the PDDI Access Software.

WORKING FORM - A memory resident form of a model that supports rapid access to entities via the Access Software.

WORKING FORMAT - The physical representation of the Working Form within the computer.

SECTION 3

SYSTEM OPERATIONS

3.1 System Overview

The purpose of the PDDI Software System is to provide a prototype for the communication of complete Production Definition Data (PDD) between dissimilar CAD/CAM Systems. This system will serve as the information interface between Engineering and Manufacturing functions. It is composed of Access Software, Conceptual Schema, Exchange Format, a Translator and Schema Manager Software (See Figure 3-1).

The Access Software is a set of callable utility programs that will allow applications to manipulate and query PDD resident in the Working Form. The Conceptual Schema is a human readable description of the data needed to define a CAD/CAM model. The Exchange Format is a physical sequential format for passing data between dissimilar systems. The Working Form is an in-core representation of a CAD/CAM model. The PDDI Translator is the software mechanism for passing this data between the Exchange Format and the Working Form. The Schema Manager is the software mechanism for managing the definitions of entities in the Working Form.

Physical Schemas

The Exchange Format physical schema is determined by the PDDI Conceptual Schema and the specification for the neutral file format. The Working Form physical schema is generated from the Conceptual Schema, with data items rearranged in order to conserve in-core memory. The content of any Working Form entity is the same as its Conceptual Schema definition; however, the order of the data items within memory may be different for storage efficiency reasons.

Software Packages

The software for the Translator system consists of three (3) packages - Access Software, Schema Manager, and Translator.

Access Software

The PDDI Access Software package is an integrated set of routines that create and manage an incore Working Form of the PDDI data structure through key access. This Access Software keeps the application independent of the actual physical definition of the Working Form. It also serves as a bridge between existing CAD/CAM systems and the PDDI Exchange Format. The PDDI Access Software reduces the task of writing the Exchange Format by providing the utility functions for initializing the Working Form model, manipulating entities, and maintaining lists.

The PDDI Access Software operates on the data structure of the application and the Working Form, by using either entity or list operations. The entity operations allow the user to create, delete, modify and query entities. List operations manage the lists which are temporary data structures containing references to entities (keys). An application can build and maintain lists specifically for its needs.

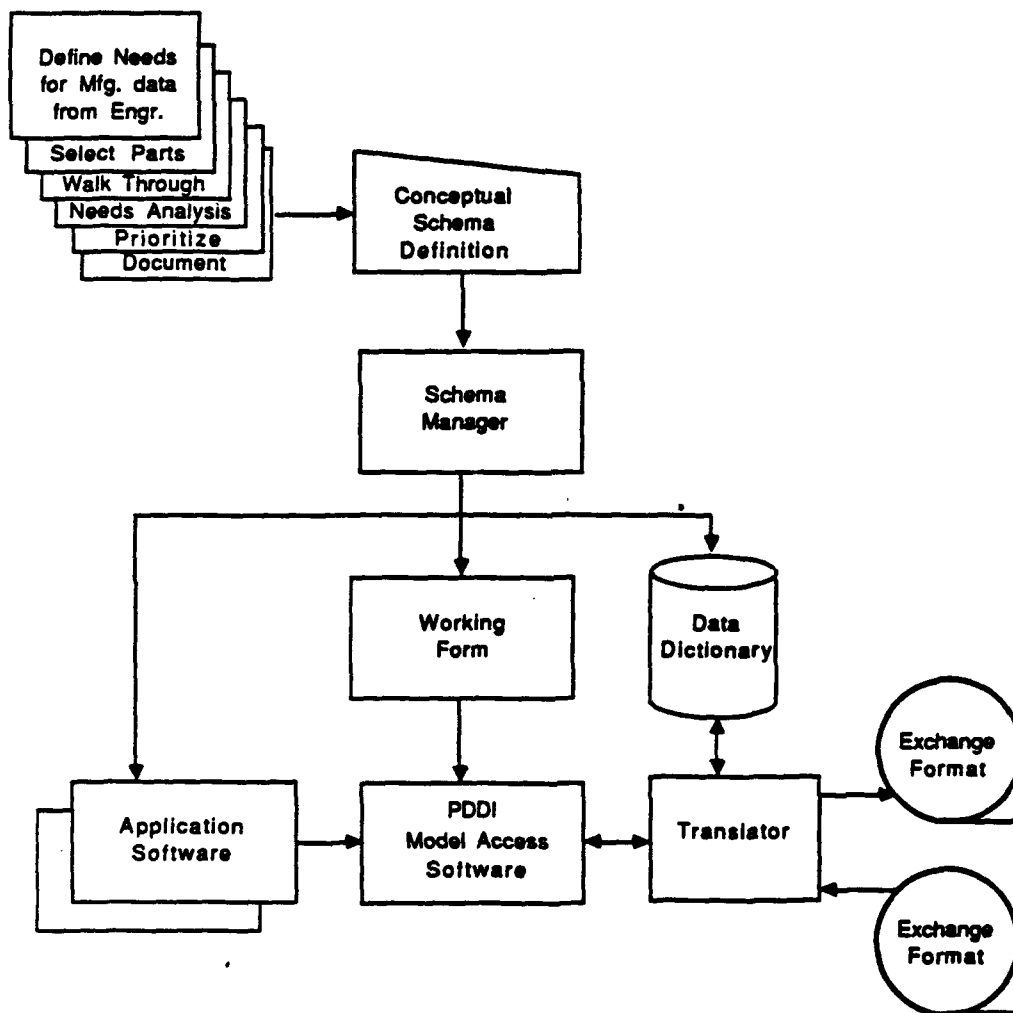


Figure 3-1 PDDI System Architecture

Schema Manager

The Schema Manager is the software package used to manage the definitions of the entities contained in the Working Form. It has three major functions:

- model a concrete conceptual schema
- transform a concrete conceptual schema into a physical schema suitable for the Working Form of the PDDI Access Software
- generate subschema forms of the physical schema for use by application programs at compile-time and/or run-time.

The Schema Manager consists of three main sub-packages: the Interactive Interface, the Batch Interface, and the Model Query Utility.

The functions of the Interactive Interface include the creation, review, update, reporting, filing, and retrieving of entity definitions. The Interactive Interface makes use of the IBM/SPF Dialog Manager for full-screen terminal menus.

The Batch Interface provides a mechanism for the creation, reporting, and filing of entity definitions in a non-interactive mode. The Batch Interface uses the syntax of the EXPRESS information modeling language (PDES/STEP) for input.

The Model Query Utility provides a mechanism for querying the (part model) entities in the Working Form. The entity definitions are used to translate the Working Form binary representation of the (part model) entities into a list of the attribute names and their values. The Model Query Utility makes use of the IBM/SPF Dialog Manager for full-screen terminal menus.

Translator

The PDDI Translator is the software package used to transmit the PDD between systems. The Translator consists of three main sub-packages. These sub-packages are: "Run System", "Pre-Processor" and "Post-Processor". (See Figure 3-2).

The Run System is the interface between the user and the "processors". Functions of this package include: Perform system configuration activities, determine files needed by the processors and make them available, and provide messages to aid user interfaces.

Access to the native database is provided for in this package via calls to a user supplied routine. The interfaces for those routines are provided in this system. The user need only write them and link them into the system. The interfaces call for PASCAL routines. If another language is used, the interfaces may need to be modified. If the pre-processor is requested, the user-supplied routine INTRTV is invoked, followed by the user supplied routine CONVRT, followed by the PDDI supplied routine PRE. INTRTV is intended to retrieve a model from the native database into the Working Form. CONVRT is intended to make any necessary conversions from the native database schema to the PDDI schema, PRE then creates an Exchange File from the Working Form built

and provided by INTRTV and CONVRT. If the post-processor is requested, the PDDI supplied routine POST is invoked, followed by the user supplied routine CONVRT, which is then followed by the user supplied routine INTFIL. POST creates a PDDI Working Form from an Exchange File. CONVRT is intended to make any necessary conversions from the PDDI schema to the native database schema. INTFIL is intended to file a model from the Working Form to the native database. To summarize, the user needs to supply three routines that provide the following functions:

INTRTV - Retrieves a model from the native database into the Working Form.

CONVRT - Converts a Working Form model from the PDDI Schema to native database schema or from native database schema to PDDI Schema depending on the choice of pre-processing or post-processing.

INTFIL - Files a model from the Working Form into the native database.

The Pre-Processor provides the interface from the Working Form to the Exchange Format.

Working Form entities, in the Working Form physical schema, are accessed via the Access Software. Data Dictionaries, obtained from the Run System Subpackage, are then used to map the Working Form entities to the Exchange Format physical schema. The Exchange Format entities are then encoded and placed into the Exchange Format file.

Transfer data is collected during entity processing. This data is encoded and placed into the Exchange Format file.

Error messages or condition codes are sent to the "Run System" to indicate the status of the transfer.

The Post-Processor provides the interface from the Exchange Format to the Working Form.

A set of tables, called data dictionaries, are obtained from the Run System Subpackage, and used to map the Exchange Format entities to the Working Form physical schema. These tables hold the physical schema of each entity in the Working Form. A description of them can be found in Appendix B of the Translator User's Manual. The Access Software is then used to place these entities into the Working Form.

3.1.1 System Interfaces

The PDDI software must interface with the computer system on which it is installed, the local (native) CAD/CAM database, the Exchange Format, the Working Form, and the user (application). It does this via the PDDI Access Software, the PDDI Translator and local (native) developed software packages. The left-hand side of Figure 3-4 shows the PDDI development environment.

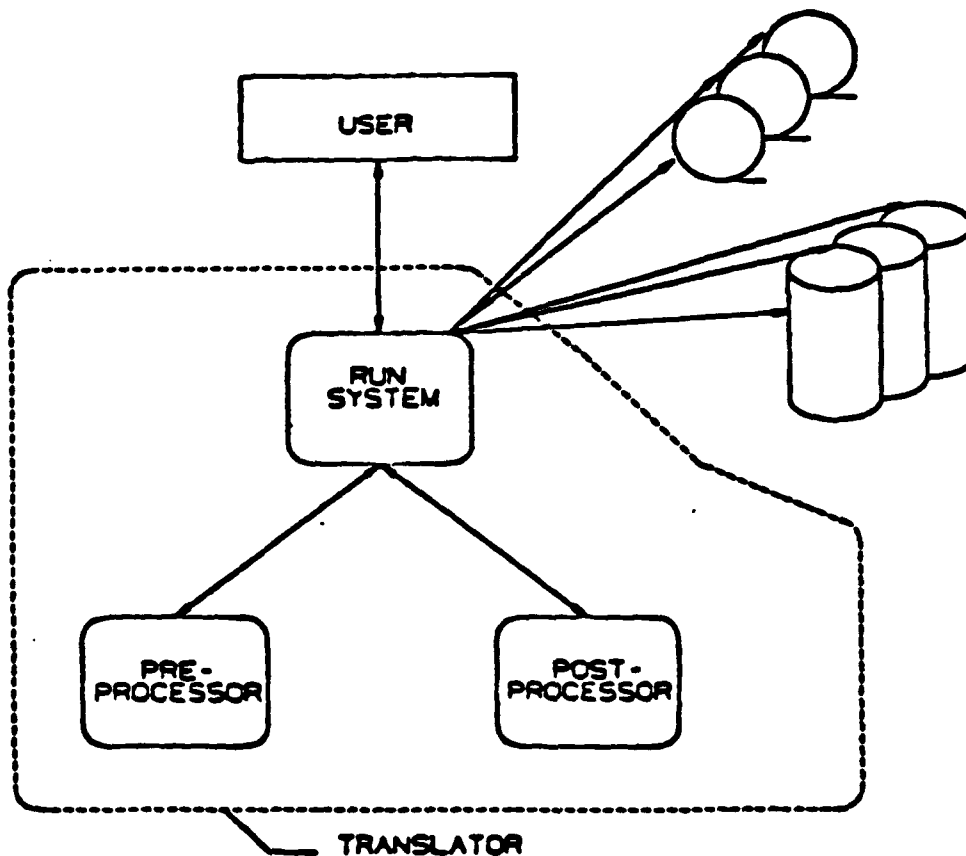


Figure 3-2 Translator Architecture

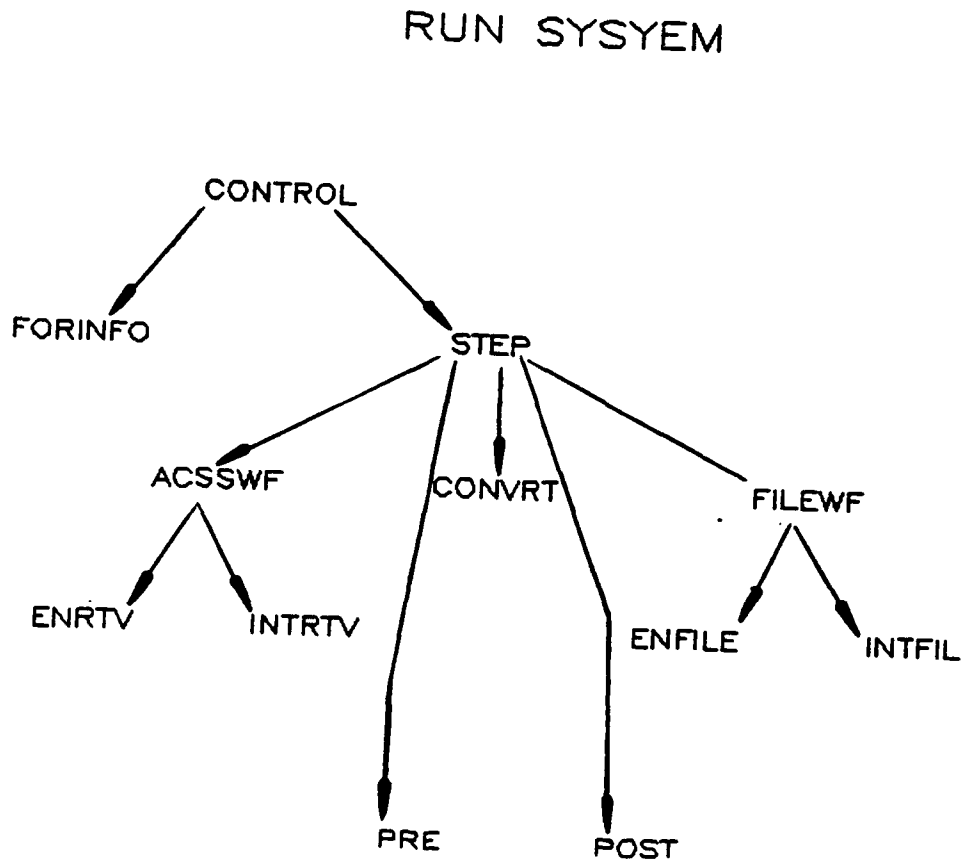


Figure 3-3 Native Database Access

3.2 System Environment

The PDDI system was developed in the following computing environment:

Computer/Operating System

IBM 43XX/MVS with TSO/MVS and associated tape drives, disk drives and terminals.

DEC VAX 11/780 VMS with associated tape drives, disk drives and terminals.

Storage (Core) Requirements

Maximum Core requirements for the PDDI software and the database is 1.0M plus the model size. The PDDI machined rib model required .57M.

Compilers

IBM-PASCAL/VS Release 2.2
DEC-PASCAL V3.5, FORTRAN 77 V4.4

Terminals

IBM 327X (or equivalent for graphic applications)
E&S PS300 (or equivalent for graphics applications)

The PDDI system is transportable to other computing systems. However, appropriate local (native) interfaces (translator) must be provided. The right-hand side of Figure 3-4 shows the PDDI commercial demonstration architecture for UNIGRAPHICS and Computervision and United Technologies Research Center (UTRC) Systems.

3.2.1 Schema Manager

The Schema Manager is the software package used to manage the definitions of the CAD/CAM entities contained in the Working Form. It produces Pascal Include Files for use by applications programs at compile time, and Data Dictionary Files for use by applications programs at run-time.

The Schema Manager Software was developed in the IBM computer system environment using the SPF Dialog Manager for full-screen terminal menus. While it is possible to migrate the Schema Manager to other computer systems, This was out-of-scope to the contracted effort. Instead, the Pascal Include files and Data Dictionary Files were migrated to the VAX computer system environment, for use by the PDDI Translator and other applications programs.

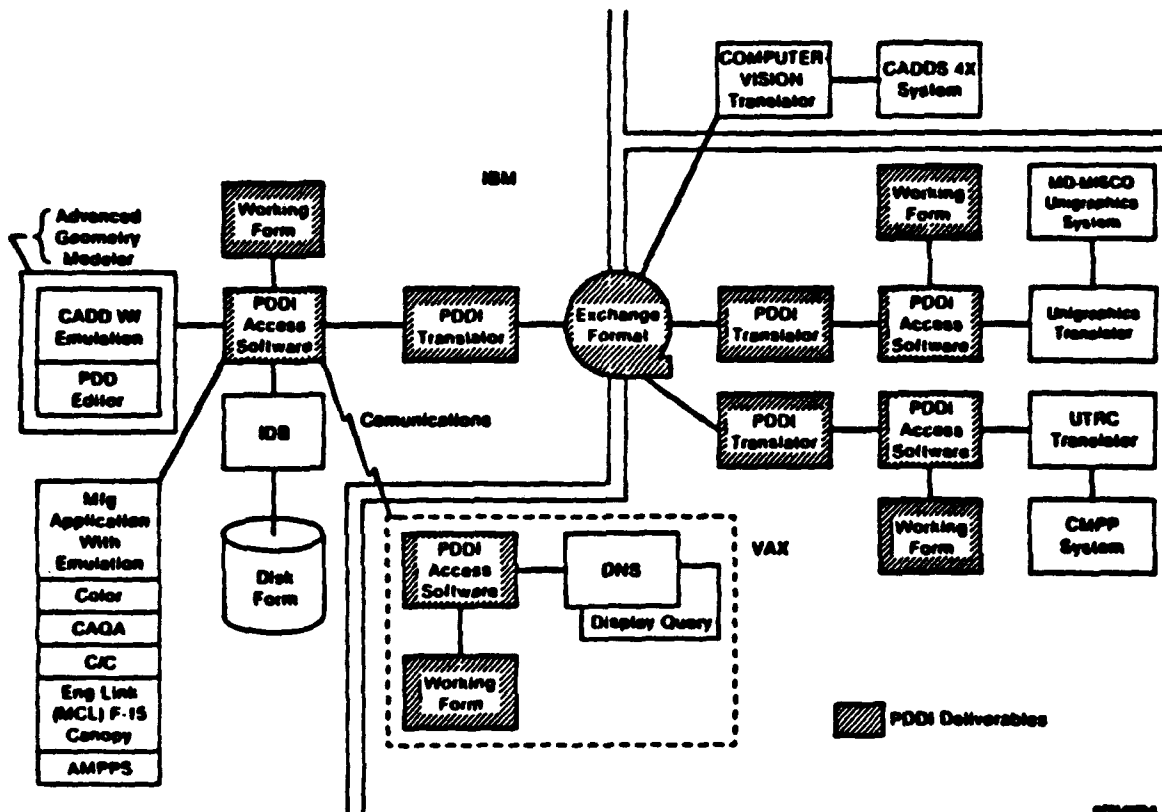


Figure 3-4 PDDI Environment

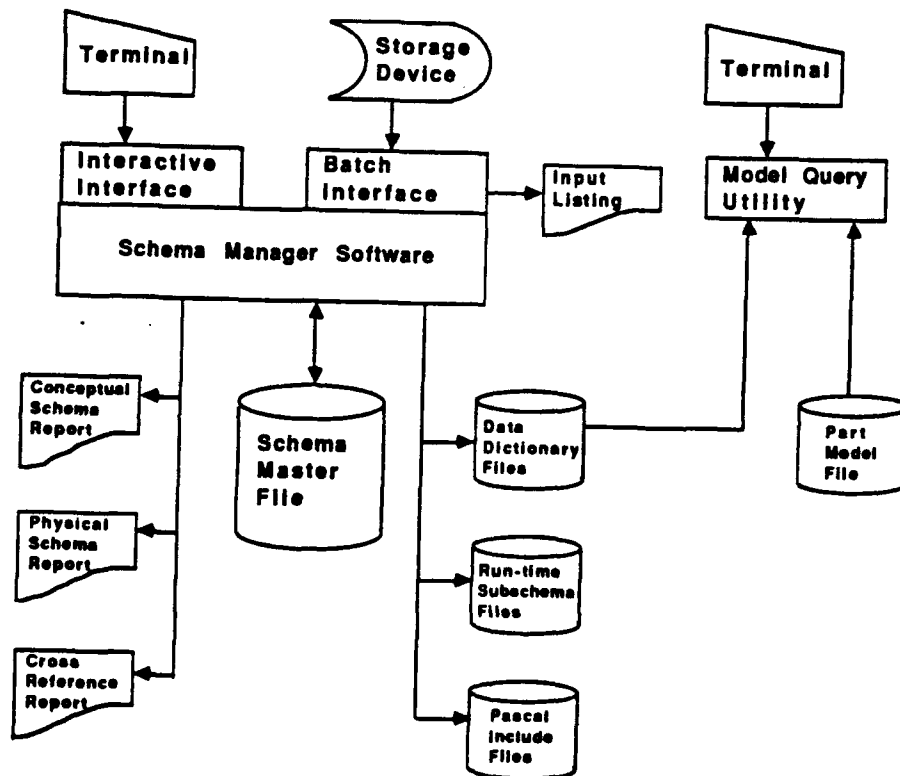


Figure 3-5 Schema Manager Architecture

SECTION 4

INSTALLATION

4.1 Overview

The PDDI Translator system uses the Exchange Format File to transfer Product Definition Data (PDD) between systems. The Exchange Format File is a neutral data file accessible by a CAD/CAM system with suitable translation capabilities.

The Exchange Format File may be translated by the receiving system, using the PDDI Translator and placed into the incore Working Form. The PDDI Access Software accepts this transformed data and puts it in the Working Form.

Conversely the incore Working Format can be accessed using the PDDI Access Software, translated using the PDDI Translator, and placed into the Exchange Format file.

A local (native) conversion utility is required to store the Working Form in the file/retrieve format of the native systems.

Conversely the native system can use this utility to retrieve the Working Form model from local (native) storage and put it in the Exchange Format using the PDDI Access Software and the PDDI Translator. Calls to these routines have been provided for in the translator software, but must be written by the user. Software that files and retrieves from a hex file has been provided. This software is intended to be an interim solution only. Local software should be written to file and retrieve from the local database.

The following descriptions and procedures apply to IBM using their terminology. Appendix B provides the IBM to VAX conversion procedures used by the PDDI development team.

4.2 Operations

4.2.1 Introduction

The exchange medium for the PDDI software is typically two (2) magnetic tapes for each computer system (IBM and VAX). One tape contains the PDDI Software and the other contains PDDI exchange format files of the PDDI parts. (See Appendix A)

4.2.2 Datasets

The PDDI system uses a collection of datasets to provide and hold data used by the system. These datasets have been assigned logical names and therefore the names must be edited to reflect native system requirements. Appendix C gives a listing of the datasets used with a description of their functions. These data files include Data Dictionaries, Message Tables, etc.

4.2.3 Operator Interface

The operator executes a series of CLISTS and JCL Command files on VAX or an IBM/MVS from a terminal on the native system to enact installation. The basic functions performed during system installation are compile and link.

4.2.4 Compile

The compile phase produces object code from source code. This object code is then used in the linkage editor. The IBM JCL and VAX Command files used to compile the PDDI Software subsystems (Translator, Access Software, PDDI INTERIM DATABASE, Schema Manager, and GET Data Dictionary (GDD)) are listed in Tables A-7 and A-8.

4.2.5 Link

The Linkage Editor produces the load module from the object code generated during the compile phase. The Pre-Processor, Post-Processor, Access Software PDDI Interim Database (PID), and GETDD (GDD) are linked together when installing the PDDI Translator. Procedures to install Access Software only for an application are provided in Paragraph 4.3.4.

4.3 Installation Procedures

4.3.1 Introduction

The IBM CLISTS and JCL and VAX Command files provided for installation are those used in the development of the PDDI prototype system. The software does not include local (native) system routines necessary to run the Translator. These are system routines supplied as a part of the computer system when delivered by the hardware vendor. The procedures provided in this manual are intended to reflect these facts and to point out where changes are needed. They do not include native system procedures. Vendor hardware and operating system manuals provide the necessary instructions for basic functions, e.g., loading tapes, native system startup, and device initialization. Other procedures needed to operate a native system should be obtained prior to PDDI system installation.

4.3.2 Preliminary Steps

- Ensure that all necessary documentation is current and available - Routine Descriptions are listed in the Product Specification.
- Ensure that appropriate personnel are familiar with the operation of the "native system".
- Ensure that adequate "native system" resources are available.
- Obtain the current PDDI Access Software Tape and PDDI Translator Tape.
- Mount and read the PDDI tapes.

4.3.3 Procedures - PDDI Software

1. Compile the source Translator, Access Software, Schema Manager, GETDD, and PID routines into object libraries.

To compile the Translator routines, use the CLIST:

```
COMTRN  
or the VAX DCL:  
COMPILE_TRN
```

To compile the Access Software routines, use the IBM JCL:

```
COMMAS
```

Refer to Paragraph 4.3.4.

To compile the Schema Manager routines, use the IBM JCL:

```
COMSCMB  
COMMQU  
COMNVI  
COMRTS  
COMSCMI
```

To compile the GETDD routine, use the IBM JCL:

```
COMGDD
```

To compile PID routines use the IBM JCL:

```
COMPID  
or the VAX DCL  
COMPILE_PID
```

2. Create and compile the routines necessary to access the native system database:

INTRTV - Retrieve from native database and place in Working Form.

CONVRT - Convert application unique entities in the Working Form to PDDI entities and vice-versa.

INTFIL - File into native database from Working Form.

Calls to access these routines are found in the Translator routines ACSSWF and FILEWF (see Figure 3-3). The declaration and parameters should be edited to match the native system software.

The object from these routines should be placed in a library for access during linkage.

A version of those routines which access the interim file and retrieve software is provided. See the Translator User's Manual.

3. When creating the Translator Load Module on IBM JCL exists to link the MAS, GDD, and PID subsystems.

This JCL is appropriately named as shown below:

```
LINKMAS  
LINKGDD  
LINKPID
```

The JCL LINK_TRN links the translator object with the MAS, GDD, and PID load modules to create the Translator Load Module.

The translator executable image on the VAX is created by executing the command file LINK_TRN. This command file includes the MAS, GDD, PID, and Translator object when creating the Translator Load Module.

4. Link shared object libraries into subsystem modules:

```
LINKNVI  
LINKRTS
```

and perform the final links (which also use the results of LINKMAS and LINKPID) into executable load modules:

```
LINKSCMB  
LINKSCMI
```

4.3.4 Procedure - Application Program reference to Access Software Only

If an application PASCAL procedure calls an Access Software interface routine, it must be compiled with a reference to the Include dataset to allow type checking of the call parameter list. This is accomplished in the IBM PASCAL/VS compiler by using the %INCLUDE facility, as follows:

1. Allocate the Include file to the compile step.
DDNAME(SYSLIB) DSN('CAD5.PDDI.MAS2.INCLD')
2. Reference the include member in the application PROCEDURE
\$INCLUDE MAxxxx where MAxxxx is the interface routine name.

To build a load module that contains the PDDI Access Software (MAS), in the IBM Linkage Editor step of the job:

1. //MASLIB DD DSN=CAD2.PDDI.MAS.LOAD, DISP=SHR
2. //SYSLIN *
INCLUDE MASLIB(MAS)

This example allocates the Access Software data set to the DDNAME of MASLIB, then allows the inclusion of the MAS load module from the allocated MASLIB.

APPENDIX A

PDDI SOFTWARE INSTALLATION

Four magnetic tapes containing the IBM and VAX versions of the PDDI deliverable software and exchange format files are available from the AF. The tapes are appropriately labeled as IBM or VAX and as software or exchange format files.

The IBM tapes were written using the IBM utility IEHMOVE (see Table A-1 and Table A-2) and are formatted as follows:

Software Files

1600 bpi
Label = PDDI
Fixed Block
Recordlength = 80
Blocksize = 800

Exchange Format Files

1600 bpi
No Label
Fixed Block
Recordlength = 80
Blocksize = 800

The VAX tapes were written using the DEC standard BACKUP utility and are formatted as follows (see Table A-3 and Table A-4).

Software and Exchange Format Files

1600 bpi
Label = PDDI
Recordlength = 80
Blocksize = 800

The datasets described in Table A-5 identify the individual PDDI software components. Detailed descriptions of the software can be found in the PDDI Product Specification.

The datasets described in Table A-6 identify the exchange format files of the models created for PDDI.

Table A-7 describes JCL and Clist files that were used to compile, link, and run the software on an IBM/MVS System at MCAIR. If the user is installing the software on IBM hardware with an MVS operating system, these files can be used with only a few minor changes (Replace MCAIR specific system libraries with your own). If the software is being installed on IBM hardware with a VM operating system, these files can only be used as a guide. There are significant changes required for use with the VM operating system.

Table A-8 describes command files that were used to compile, link and run the software on a VAX/VMS System at MCAIR. If the user is installing the software on DEC hardware with a VMS operating system, these files can be used without any changes to them.

TABLE A-1

PDDI SOFTWARE TAPE JCL
IBM/MVS

```
//TM360JPT JOB ACCNTNO,'PDDI',NOTIFY=TM360JP,
//      REGION=3000K,MSGCLASS=T,LIM=(30,90,60,60)
//*FORMAT PR,DDNAME,=DEST=N002
//*MAIN CLASS=A
//S1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(8))
//TAPE1 DD UNIT=TAPE62,DISP=(,PASS),LABEL=(,SL),DCB=DEN=3,
//      VOL=SER=(PDDI)
//DD1 DD UNIT=3380,DISP=SHR,VOL=SER=LH02DO
//SYSIN DD *
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,01),      X
      DSNAME=CAD5.PDDI.V30.CLIST
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,02),      X
      DSNAME=CAD5.PDDI.V30.CNTL
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,03),      X
      DSNAME=CAD5.PDDI.V30.DATA
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,04),      X
      DSNAME=CAD5.PDDI.V30.DDDEFN.DATA
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,05),      X
      DSNAME=CAD5.PDDI.V30.DDDTIP.DATA
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,06),      X
      DSNAME=CAD5.PDDI.V30.DDINDX.DATA
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,07),      X
      DSNAME=CAD5.PDDI.V30.DDITIP.DATA
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,08),      X
      DSNAME=CAD5.PDDI.V30.GDDINC
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,09),      X
      DSNAME=CAD5.PDDI.V30.GDDSRC
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,10),      X
      DSNAME=CAD5.PDDI.V30.MASINC
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,11),      X
      DSNAME=CAD5.PDDI.V30.MASSRC
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,12),      X
      DSNAME=CAD5.PDDI.V30.MQUINC
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,13),      X
      DSNAME=CAD5.PDDI.V30.MQMSG
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,14),      X
      DSNAME=CAD5.PDDI.V30.MQUPAN
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,15),      X
      DSNAME=CAD5.PDDI.V30.MQUSRC
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,16),      X
      DSNAME=CAD5.PDDI.V30.NVIINC
COPY TODD=TAPE1,FROM=3380=LH02DO,TO=TAPE62=(PDDI,17),      X
      DSNAME=CAD5.PDDI.V30.NVISRC
```

OM 5601300008
22 December 1987

COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 18), DSNAME=CAD5.PDDI.V30.PIDINC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 19), DSNAME=CAD5.PDDI.V30.PIDSRC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 20), DSNAME=CAD5.PDDI.V30.RTSINC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 21), DSNAME=CAD5.PDDI.V30.RTSSRC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 22), DSNAME=CAD5.PDDI.V30.SCMBINC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 23), DSNAME=CAD5.PDDI.V30.SCMSRC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 24), DSNAME=CAD5.PDDI.V30.SCMINC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 25), DSNAME=CAD5.PDDI.V30.SCMMSG	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 26), DSNAME=CAD5.PDDI.V30.SCMIPAN	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 27), DSNAME=CAD5.PDDI.V30.SCMISRC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 28), DSNAME=CAD5.PDDI.V30.TIPINC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 29), DSNAME=CAD5.PDDI.V30.TIPMAP	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 30), DSNAME=CAD5.PDDI.V30.TIPSRC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 31), DSNAME=CAD5.PDDI.V30.TRNDAT	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 32), DSNAME=CAD5.PDDI.V30.TRNINC	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 33), DSNAME=CAD5.PDDI.V30.TRNMSG	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 34), DSNAME=CAD5.PDDI.V30.TRNPAN	X
COPY TODD=TAPE1, FROM=3380=LH02DO, TO=TAPE62=(PDDI, 35), DSNAME=CAD5.PDDI.V30.TRNSRC	X

/*
//

TABLE A-2

PDDI EXCHANGE FORMAT FILES TAPE JCL
IBM/MVS

```
//TM360JPT JOB ACCNTNO,'PDDI',NOTIFY=TM360JP,
//      REGION=3000K,MSGCLASS=T,LIM=(30,90,60,60)
//*FORMAT PR,DDNAME,=DEST=N002
//*MAIN CLASS=A
//S1 EXEC PGM=IFHMOVE
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(8))
//TAPE1 DD UNIT=TAPE62,DISP=(,PASS),LABEL=(,NL),DCB=DEN=3,
//      VOL=SER=(PDDI)
//DD1 DD UNIT=3380,DISP=SHR,VOL=SER=LH02D0
//SYSIN DD *
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,01),      X
      DSN=NAME=CAD5.PDDI.V30.EF.B1SC.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,02),      X
      DSN=NAME=CAD5.PDDI.V30.EF.COMRIB.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,03),      X
      DSN=NAME=CAD5.PDDI.V30.EF.ELECTRIC.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,04),      X
      DSN=NAME=CAD5.PDDI.V30.EF.MACRIB3.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,05),      X
      DSN=NAME=CAD5.PDDI.V30.EF.MACRIB5.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,06),      X
      DSN=NAME=CAD5.PDDI.V30.EF.SMRIB.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,07),      X
      DSN=NAME=CAD5.PDDI.V30.EF.TIPIGES.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,08),      X
      DSN=NAME=CAD5.PDDI.V30.EF.TIPPDES.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,09),      X
      DSN=NAME=CAD5.PDDI.V30.EF.TSTMOD.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PDDI,10),      X
      DSN=NAME=CAD5.PDDI.V30.EF.TURNPT.DATA
/*
//
```


TABLE A-3

PDDI SOFTWARE TAPE DCL
VAX/VM

```
$ SET VERIFY
$!
$! THIS WILL CREATE A TAPE OF THE PDDI SYSTEM SOFTWARE
$!
$ DELETE TAPE_LIST.LST;*
$ INIT/DENSITY=1600 MUA0: PDDI
$ MOUNT/FOREIGN MUA0: PDDI
$
$! COPY COMMAND PROCEDURES FOR COMPILING, LINKING, AND RUNNING
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
[PDDI.V30.COMFIL]ASSIGN_DIR.COM;;-
[PDDI.V30.COMFIL]ASSIGN_TRN_BATCH.COM;;-
[PDDI.V30.COMFIL]ASSIGN_TRN_INTER.COM;;-
[PDDI.V30.COMFIL]COMPILE_GDD.COM;;-
[PDDI.V30.COMFIL]COMPILE_MAS.COM;;-
[PDDI.V30.COMFIL]COMPILE_PID.COM;;-
[PDDI.V30.COMFIL]COMPILE_TRN.COM;;-
[PDDI.V30.COMFIL]INC_ASSIGN_GDD.COM;;-
[PDDI.V30.COMFIL]INC_ASSIGN_MAS.COM;;-
[PDDI.V30.COMFIL]INC_ASSIGN_PID.COM;;-
[PDDI.V30.COMFIL]INC_ASSIGN_TRN.COM;;-
[PDDI.V30.COMFIL]RUN_TRN_BATCH.COM;;-
[PDDI.V30.COMFIL]RUN_TRN_INTERACTIVE.COM;;-
[PDDI.V30.COMFIL]TRN_LINK.COM -
MUA0:COMTIL.BKP/SAVESET
$
$! COPY DATA DICTIONARY AND PASCAL INCLUDE FILES
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
[PDDI.V30.DDFILS]PDDI_DDD.DAT;;-
[PDDI.V30.DDFILS]PDDI_DDI.DAT;;-
[PDDI.V30.DDFILS]MSGTBL2.DAT -
MUA0:DDFILS.BKP/SAVESET
$
$! COPY GETDD INCLUDE FILES
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
[PDDI.V30.GDDINC]*.* -
MUA0:GDDINC.BKP/SAVESET
$
$! COPY GETDD SOURCE FILES
$
```

```
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -  
  [PDDI.V30.GDDSRC]*.* -  
  MUA0:GDDSRC.BKP/SAVESET  
$  
$! COPY MAS INCLUDE FILES  
$  
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -  
  [PDDI.V30.MASINC]*.* -  
  MUA0:MASINC.BKP/SAVESET  
$  
$! COPY MAS SOURCE FILES  
$  
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -  
  [PDDI.V30.MASSRC]*.* -  
  MUA0:MASSRC.BKP/SAVESET  
$  
$! COPY PID INCLUDE FILES  
$  
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -  
  [PDDI.V30.PIDINC]*.* -  
  MUA0:PIDINC.BKP/SAVESET  
$  
$! COPY PID SOURCE FILES  
$  
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -  
  [PDDI.V30.PIDSRC]*.* -  
  MUA0:PIDSRC.BKP/SAVESET  
$  
$! COPY TRAN INCLUDE FILES  
$  
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -  
  [PDDI.V30.TRNINC]*.* -  
  MUA0:TRNINC.BKP/SAVESET  
$  
$! COPY TRAN SOURCE FILES  
$  
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -  
  [PDDI.V30.TRNSRC]*.* -  
  MUA0:TRNSRC.BKP/SAVESET  
$  
$ DISMOUNT/NOUNLOAD MUA0:  
$ COPY/CONCATENATE TAPE_LIST.LST;* PDDI_V30_TAPE.LST  
$ SET NOVERIFY  
$ EXIT
```

TABLE A-4

PDDI EXCHANGE FORMAT FILES TAPE DCI
VAX/VM

```
$ SET VERIFY
$!
$! THIS WILL CREATE A TAPE OF THE EXCHANGE FORMATS FOR THE PDDI MODELS
$!
$ DELETE TAPE_LIST.LST;*
$ INIT/DENSITY=1600 MUA0: PDDI
$ MOUNT/FOREIGN MUA0: PDDI
$
$! COPY COMMAND PROCEDURES FOR COMPILING, LINKING, AND RUNNING
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
  [PDDI.V30.EFFILS]BISC.DAT;,-
  [PDDI.V30.EFFILS]COMRIB.DAT;,-
  [PDDI.V30.EFFILS]ELEC.DAT;,-
  [PDDI.V30.EFFILS]MACRIB3.DAT;,-
  [PDDI.V30.EFFILS]MACRIB5.DAT;,-
  [PDDI.V30.EFFILS]SMRIB.DAT;,-
  [PDDI.V30.EFFILS]TSTMOD.DAT;,-
  [PDDI.V30.EFFILS]TURNPT.DAT -
  MUA0:EFFILS.BKP/SAVESET
$ DISMOUNT/NOUNLOAD MUA0:
$ COPY/CONCATENATE TAPE_LIST.LST;* PDDI_V30_EF.LST
$ SET NOVERIFY
$ EXIT
```

TABLE A-5
PDDI SOFTWARE COMPONENTS

<u>File Name</u>	<u>Hardware</u>	<u>Description</u>
[PDDI.V30.GDDINC]	VAX	Contains GETDD(GDD) pascal include files. This routine is used to read the PDDI data dictionary.
CAD5.PDDI.V30.GDDINC	IBM	
[PDDI.V30.GDDSRC]	VAX	Contains GETDD(GDD) source files. This routine is used to read the PDDI data dictionary.
CAD5.PDDI.V30.GDDSRC	IBM	
PDDI.V30.MASINC]	VAX	Contains Model Access Software (MAS) pascal include files.
CAD5.PDDI.V30.MASINC	IBM	
[PDDI.V30.MASSRC]	VAX	Contains Model Access Software (MAS) source files.
CAD5.PDDI.V30.MASSRC	IBM	
CAD5.PDDI.V30.MQUINC	IBM	Contains Model Query Utility (MQU) pascal include files.
CAD5.PDDI.V30.MQUSRC	IBM	
CAD5.PDDI.V30.MQUPAN	IBM	Contains Model Query Utility (MQU) SPF panels. Contains Model Query Utility (MQU) SPF Panel message libraries.
CAD5.PDDI.V30.MQUMSG	IBM	
[PDDI.V30.NVIINC]	VAX	Contains Name/Value Interface (NVI) pascal include files.
CAD5.PDDI.V30.NVIINC	IBM	
[PDDI.V30.NVISRC]	VAX	Contains Name/Value Interface (NVI) source files.
CAD5.PDDI.V30.NVISRC	IBM	
[PDDI.V30.PIDINC]	VAX	Contains PDDI Interim Database (PID) pascal include files.
CAD5.PDDI.V30.PIDINC	IBM	
[PDDI.V30.PIDSRC]	VAX	Contains PDDI Interim Database (PID) source files.
CAD5.PDDI.V30.PIDSRC	IBM	

<u>File Name</u>	<u>Hardware</u>	<u>Description</u>
CAD5.PDDI.V30.RTSINC	IBM	Contains Run-Time Subschema (RTS) pascal include files.
CAD5.PDDI.V30.RTSSRC	IBM	Contains Run-Time Subschema (RTS) source files.
CAD5.PDDI.V30.SCMBINC	IBM	Contains Batch Schema Manager (SCMB) pascal include files.
CAD5.PDDI.V30.SCMSRC	IBM	Contains Batch Schema Manager (SCMB) source files.
CAD5.PDDI.V30.SCMINC	IBM	Contains Interactive Schema Manager (SCMI) pascal include files.
CAD5.PDDI.V30.SCMISRC	IBM	Contains Interactive Schema Manager (SCMI) source files.
CAD5.PDDI.V30.SCMIPAN	IBM	Contains Interactive Schema Manager (SCMI) SPF panels.
CAD5.PDDI.V30.SCMIMSG	IBM	Contains Interactive Schema Manager (SCMI) message libraries.
CAD5.PDDI.V30.TIPINC	IBM	Contains IGES/PDES Translator (TIP) pascal include files.
CAD5.PDDI.V30.TIPSRC	IBM	Contains IGES/PDES Translator (TIP) source files.
CAD5.PDDI.V30.TIPMAP	IBM	Contains IGES/PDES Translator (TIP) mapping file.
[PDDI.V30.TRNINC]	VAX	Contains PDDI System Translator (TRN) pascal include files.
CAD5.PDDI.V30.TRNINC	IBM	
[PDDI.V30.TRNSRC]	VAX	Contains PDDI System Translator (TRN) source files.
CAD5.PDDI.V30.TRNSRC	IBM	
CAD5.PDDI.V30.TRNPAN	IBM	Contains PDDI System Translator (TRN) SPF panels.
CAD5.PDDI.V30.TRNMSG	IBM	Contains PDDI System Translator (TRN) SPF Panel message libraries.
CAD5.PDDI.V30.TRNDAT	IBM	Contains PDDI System Translator (TRN) data files.
CAD5.PDDI.V30.DDDEFN.DATA	IBM	PDDI Data dictionary.
CAD5.PDDI.V30.DDINDX.DATA	IBM	PDDI Data dictionary index file.

<u>File Name</u>	<u>Hardware</u>	<u>Description</u>
[PDDI.V30.DDFILS]	VAX	Contains data dictionary, data dictionary index file, and pascal include files that describe the PDDI entity definitions.
CAD5.PDDI.V30.DDDTIP.DATA	IBM	Contains IGES/PDES Translator (TIP) data dictionary.
CAD5.PDDI.V30.DDITIP.DATA	IBM	Contains IGES/PDES Translator (TIP) data dictionary index file.

TABLE A-6
PDDI EXCHANGE FORMAT FILES

<u>File Name</u>	<u>Hardware</u>	<u>Description</u>
[PDDI.V30.EFFILS]BISC.DAT	VAX	Contains PDDI Exchange Format File for the B1 Spar Clip.
CAD5.PDDI.V30.EF.BISC.DATA	IBM	
[PDDI.V30.EFFILS]COMRIB.DAT	VAX	Contains PDDI Exchange Format File for the Composite Rib.
CAD5.PDDI.V30.EF.COMRIB.DATA	IBM	
[PDDI.V30.EFFILS]ELECTRIC.DAT	VAX	Contains PDDI Exchange Format File for the Electronic Part.
CAD5.PDDI.V30.EF.ELECTRIC.DATA	IBM	
[PDDI.V30.EFFILS]MACRIB3.DAT	VAX	Contains PDDI Exchange Format File for the 3-Axis Machined Rib.
CAD5.PDDI.V30.EF.MACRIB3.DATA	IBM	
[PDDI.V30.EFFILS]MACRIB5.DAT	VAX	Contains PDDI Exchange Format File for the 5-Axis Machined Rib.
CAD5.PDDI.V30.EF.MACRIB5.DATA	IBM	
[PDDI.V30.EFFILS]SMRIB.DAT	VAX	Contains PDDI Exchange Format File for the Sheet Metal Rib.
CAD5.PDDI.V30.EF.SMRIB.DATA	IBM	
CAD5.PDDI.V30.EF.TIPIGES.DATA	IBM	Contains an IGES Exchange Format File used in the IGES/PDES Translator.
CAD5.PDDI.V30.EF.TIPPDES.DATA	IBM	Contains a PDES Exchange Format File created by the IGES/PDES Translator.
[PDDI.V30.EFFILS]TSTMOD.DAT	VAX	Contains PDDI Exchange Format File for the Test Model.
CAD5.PDDI.V30.EF.TSTMOD.DATA	IBM	

OM 560130000B
22 December 1987

<u>File Name</u>	<u>Hardware</u>	<u>Description</u>
[PDDI.V30.EFFILS]TURNPT.DAT	VAX	Contains PDDI Exchange Format File for the Turned Part.
CAD5.PDDI.V30.EF.TURNPT.DATA	IBM	

TABLE A-7
PDDI IBM JCL AND CLISTS

<u>File Name</u>	<u>Description</u>
CAD5.PDDI.V30.CNTL(BATPRE)	IBM/MVS JCL to submit a batch pre-processor translation.
CAD5.PDDI.V30.CNTL(BATPST)	IBM/MVS JCL to submit a batch post-processor translation
CAD5.PDDI.V30.CNTL(COMGDD)	IBM/MVS JCL to submit a batch compilation of GETDD(GDD) source.
CAD5.PDDI.V30.CNTL(COMMAS)	IBM/MVS JCL to submit a batch compilation of Model Access Software (MAS) source.
CAD5.PDDI.V30.CNTL(COMMQU)	IBM/MVS JCL to submit a batch compilation of Model Query Utility (MQU) source.
CAD5.PDDI.V30.CNTL(COMNVI)	IBM/MVS JCL to submit a batch compilation of Name/Value Interface (NVI) source.
CAD5.PDDI.V30.CNTL(COMPID)	IBM/MVS JCL to submit a batch compilation of PDDI Interim Database (PID) source.
CAD5.PDDI.V30.CNTL(COMRTS)	IBM/MVS JCL to submit a batch compilation of Run-Time Subschema (RTS) source.
CAD5.PDDI.V30.CNTL(COMSCMB)	IBM/MVS JCL to submit a batch compilation of Batch Schema Manager (SCMB) source.
CAD5.PDDI.V30.CNTL(COMSCMI)	IBM/MVS JCL to submit a batch compilation of Interactive Schema Manager (SCMI) source.
CAD5.PDDI.V30.CNTL(COMTIP)	IBM/MVS JCL to submit a batch compilation of IGES/PDES Translator (TIP) source.

<u>File Name</u>	<u>Description</u>
CAD5.PDDI.V30.CNTL(COMTRN)	IBM/MVS JCL to submit a batch compilation of PDDI System Translator (TRN) source.
CAD5.PDDI.V30.CNTL(LINKGDD)	IBM/MVS JCL to submit a batch link of the GETDD(GDD) load module.
CAD5.PDDI.V30.CNTL(LINKMAS)	IBM/MVS JCL to submit a batch link of the Model Access Software (MAS) load module.
CAD5.PDDI.V30.CNTL(LINKMQU)	IBM/MVS JCL to submit a batch link of the Model Query Utility (MQU) load module.
CAD5.PDDI.V30.CNTL(LINKNVI)	IBM/MVS JCL to submit a batch link of the Name/Value Interface (NVI) load module.
CAD5.PDDI.V30.CNTL(LINKPID)	IBM/MVS JCL to submit a batch link of the PDDI Interim Database (PID) load module.
CAD5.PDDI.V30.CNTL(LINKRTS)	IBM/MVS JCL to submit a batch link of the Run-Time Subschema (RTS) load module.
CAD5.PDDI.V30.CNTL(LINKSCMB)	IBM/MVS JCL to submit a batch link of the Batch Schema Manager (SCMB) load module.
CAD5.PDDI.V30.CNTL(LINKSCMI)	IBM/MVS JCL to submit a batch link of the Interactive Schema Manager (SCMI) load module.
CAD5.PDDI.V30.CNTL(LINKTIP)	IBM/MVS JCL to submit a batch link of the IGES/PDES Translator (TIP) load module.
CAD5.PDDI.V30.CNTL(LINKTRN)	IBM/MVS JCL to submit a batch link of the PDDI System Translator (TRN) load module.
CAD5.PDDI.V30.CLIST(RUNMQU)	IBM/MVS Clist to execute the Model Query Utility (MQU) module.

<u>File Name</u>	<u>Description</u>
CAD5.PDDI.V30.CLIST(RUNSCHM)	IBM/MVS Clist to execute both the Interactive Schema Manager (SCMI) and the Batch Schema Manager (SCMB) module.
CAD5.PDDI.V30.CLIST(RUNTIP)	IBM/MVS Clist to execute the IGES/PDES Translator (TIP) module.
CAD5.PDDI.V30.CLIST(HEADERTP)	IBM/MVS Clist used to write the header section of an Exchange Format File created by the IGES/PDES Translator (TIP).
CAD5.PDDI.V30.CLIST(RUNTRAN)	IBM/MVS Clist to execute the PDDI System Translator (TRN) module.
CAD5.PDDI.V30.CLIST(FSTAT)	IBM/MVS Clist used by the PDDI System Translator (TRN) for file allocation.
CAD5.PDDI.V30.CLIST(HEADER)	IBM/MVS Clist used to write the header section of a PDDI Exchange Format File created by the PDDI System Translator (TRN).
CAD5.PDDI.V30.CLIST(NUPSTPRE)	IBM/MVS Clist used to submit batch jobs of the PDDI System Translator (TRN) from SPF Panels.
CAD5.PDDI.V30.CLIST(PDESPAN)	IBM/MVS Clist used to allocate PDDI System Translator (TRN) files for SPF.
CAD5.PDDI.V30.CLIST(PDESPANL)	IBM/MVS Clist used to allocate the SPF files for the PDDI System Translator (TRN).
CAD5.PDDI.V30.CLIST(POSTPRE)	IBM/MVS Clist used to submit batch jobs of the PDDI System Translator (TRN) from a clist.
CAD5.PDDI.V30.CLIST(MQCKDDD, MQCKDDI, MQCKDSN, MQCKOUT)	IBM/MVS Clists to support the execution of the Model Query Utility (MQU).

TABLE A-8

PDDI VAX COMMAND FILES

<u>File Name</u>	<u>Description</u>
[PDDI.V30.COMFIL]COMPILE_GDD.COM	VAX/VMS Command File to compile GETDD (GDD) source.
[PDDI.V30.COMFIL]COMPILE_MAS.COM	VAX/VMS Command File to compile Model Access Software (MAS) source.
[PDDI.V30.COMFIL]COMPILE_PID.COM	VAX/VMS Command File to compile PDDI Interim Database (PID) source.
[PDDI.V30.COMFIL]COMPILE_TRN.COM	VAX/VMS Command File to compile PDDI System Translator (TRN) source.
[PDDI.V30.COMFIL]TRN_LINK.COM	VAX/VMS Command File to link the PDDI System Translator (TRN).
[PDDI.V30.COMFIL]RUN_TRN_BATCH.COM	VAX/VMS Command File to run a batch session of the System Translator (TRN).
[PDDI.V30.COMFIL] RUN_TRN_INTERACTIVE.COM	VAX/VMS Command File to run an interactive session of the PDDI System Translator (TRN).
[PDDI.V30.COMFIL]ASSIGN_DIR.COM	VAX/VMS Command File to assign PDDI directory names to logical names.
[PDDI.V30.COMFIL] ASSIGN_TRN_BATCH.COM	VAX/VMS Command File to assign external files needed to run the PDDI System Translator (TRN) in batch mode.
[PDDI.V30.COMFIL] ASSIGN_TRN_INTER.COM	VAX/VMS Command File to assign external files needed to run the PDDI System Translator (TRN) interactively.
[PDDI.V30.COMFIL]INC_ASSIGN_GDD.COM	VAX/VMS Command File to assign the GETDD(GDD) pascal include files to logical names.
[PDDI.V30.COMFIL]INC_ASSIGN_MAS.COM	VAX/VMS Command File to assign the Model Access Software (MAS) pascal include files to logical names.

OM 560130000B
22 Decembe~ 1987

<u>File Name</u>	<u>Description</u>
[PDDI.V30.COMFIL]INC_ASSIGN_PID.COM	VAX/VMS Command File to assign the PDDI Interim Database (PID) pascal include files to logical names.
[PDDI.V30.COMFIL]INC_ASSIGN_TRN.COM	VAX/VMS Command File to assign the PDDI System Translator (TRN) pascal include files to logical names.

APPENDIX B

PASCAL (EXTENDED) IMPLEMENTATION DIFFERENCES

IBM	VAX
1. SEGMENT	! 1. MODULE
2. %INCLUDE MEMBER OF PDS	! 2. %INCLUDE 'VAXDSN'
3. %PRINT ON	! 3. %INCLUDE 'VAXDSN/LIST' (DEFAULT)
4. %PRINT OFF	! 4. %INCLUDE 'VAXDSN/NOLIST'
5. %PAGE	! 5. PAGE(INPUT OR OUTPUT)
6. @ POINTER VARIABLE SYMBOL	! 6. ^ POINTER VARIABLE SYMBOL
7. STRING IS A PREDECLARED IDENTIFIER	! 7. STRING IS NOT A PREDECLARED IDENTIFIER
8. PACKED 1..65 IS LEGAL	! 8. PACKED 1..65 IS NOT LEGAL
9. CONST IN A PROCEDURE DECLARATION e>g> PROCEDURE PROCNAME(CONST I:INTEGER);	! 9. USE [READONLY] ATTRIBUTE e.g. XINEGER=[READONLY] INTEGER; PROCEDURE PROCNAME(I:INTEGER);
10. LABEL CAN BE A VARIABLE NAME e.g. EXIT OUT:	! 10. LABEL HAS TO BE MADE UP OF DECIMAL DIGITS e.g. 10:
11. // CONCATENATION SYMBOL	! 11. + CONCATENATION SYMBOL
12. VALUE - USED FOR SPECIFYING INITIAL VALUES FOR STATIC AND DEF VAR.	! 12. VALUE - INITIAL VALUES FOR ORDINAL, REAL, AND STRUCTURAL VAR. (EXCEPT FILE VAR.), WITH CONSTANTS OR CONSTRUCTORS OF THE SAME TYPE. CAN ONLY APPEAR IN THE MAIN PROGRAM.
13. LENGTH(X)	! 13. LENGTH(X)
14. MAXLENGTH(X)	! 14. LENGTH(X)
15. DELETE FUNCTION FOR STRING	! 15. USE PAD & SUBSTRING FUNCTION TO SIMULATE DELETE ON IBM
16. S2:-S1 WHERE S2 & S1 ARE PACKED ARRAY[1..N] OF CHAR 7 S2.GE.S1 - IS LEGAL	! 16. NOT LEGAL, TO MAKE IT LEGAL S2 & S1 SHOULD BE DECLARED VARYING[N] OF CHAR
17. MIN & MAX FUNCTION	! 17. CONSTRUCT YOUR OWN MIN & MAX
18. LTRIM FUNCTION	! 18. USE LIB\$SPKC,PAD,SUBSTR & LENGTH FUNCTION TO SIMULATE LTRIM ON IBM
19. READSTR FUNCTION	! 19. READV FUNCTION

PASCAL (EXTENDED)
IMPLEMENTATION DIFFERENCES

<u>IBM</u>	<u>VAX</u>
20. PASCAL ALLOWS YOU TO USE A BASE TYPE IDENTIFIER IN A POINTER TYPE DEFINITION BEFORE YOU DEFINE THE BASE TYPE. THE BASE TYPE DOES NOT NEED TO BE DEFINED BEFORE THE END OF TYPE SECTION IN WHICH IT WAS FIRST REFERENCED.	! 20. SAME AS IBM. EXCEPT, THE BASE TYPE MUST BE DEFINED BEFORE THE END OF TYPE SECTION IN WHICH IT WAS FIRST REFERENCED. ! ! ! ! ! !
21. SUBPROGRAM	! 21. EXTERNAL
22. REF	! 22. VAR
DEF	! X : [EXTERNAL]X_TYPE
X : X_TYPE	!
23. numberX hexadecimal	! 23. %Xnumber hexadecimal
24. & Binary Logical AND	! 24. UAND Binary Logical AND

APPENDIX C

DATASETS USED BY PDDI

The following datasets are used by the Translator. Each dataset has an associated unit name (ddname) by which the Translator uses to read from/write to the dataset. The IBM CLISTS & JCL provided with the translator allocates these files and puts the appropriate values in them if necessary. Any file with a time stamp is generated by the CLIST or by the Translator. The files DBASE, EFFILE (if post-processing), EXPLAN (if pre-processing), HEADER (if pre-processing), and PASFIL are generated by the CLIST before translation begins. A description of the necessary values is provided in this Appendix. The files EFFILE (if pre-processing), FT08F001 (if post-processing), KEYFILE (if post-processing), MAPFIL, and OUTFIL (if generating a Working Form print out) are generated during the translation.

DDNAME : DBASE

Data Set Name : &USER_ID.D&TIME.RECORD

Used By : PRE, POST

Description:

Contains a parameter which indicates whether the user wants to use the PDDI supplied database software, or their internal database for file/retrieval. If the user wants their internal database, the first (and only) record contains a character 'I'. If the user wants the PDDI supplied database, the record contains a character 'P'.

DDNAME : DDFILE

Data Set Name : CAD5.PDDI.V30.DDDEFN.DATA

Used By: PRE, POST, Working Form Printout

Description:

Contains the data dictionary definitions. A description of the data dictionaries can be found in the PDDI Translator User's Manual.

DDNAME : DDINX

Data Set Name : CAD5.PDDI.V30.DDINDX.DATA

Used By : PRE, POST, Working Form Printout

Description:

Contains the data dictionary index. A description of the data dictionaries can be found in the PDDI Translator User's Manual.

DDNAME : EFFILE

Data Set Name : &USER_ID.T&TIME.EF

Used By : PRE, POST

Description:

The Exchange Format File that is created or processed by the Translator.

DDNAME : EXPLAN

Data Set Name : &USER_ID.E&TIME.EXPLAN

Used By : PRE

Description:

Contains the explanation records to go in the Exchange Format File. These will be read in and exist in the Exchange File as comments.

DDNAME : FT08F001

Data Set Name : Provided By User

Used By : PRE, POST

Description:

Contains the PDDI supplied database of the Working Form. A description of this file is provided in the PDDI Translator User's Manual, Appendix D.

DDNAME : HEADER

Data Set Name : &USER_ID.H&TIME.FILE

Used By : PRE

Description:

Contains information for header section in Exchange File. The header file is formatted as follows:

<u>Record Number</u>	<u>Information</u>
1	Name of Exchange File
2	Date in YYYYMMDD.HHMMSS format
3	User's name
4	User's organization

DDNAME : KEYFILE

Data Set Name : &USER_ID.K&TIME.KEYFILE

Used By : POST

Description:

Contains the MDC database key corresponding to the Exchange Format File after post-processing.

DDNAME : MAPFIL

Data Set Name : &USER_ID.M&TIME.MAP

Used By : PRE, POST

Description:

Contains the correspondence between the Exchange Format identifier and the Working Form entity's kind and ident. This file is intended for testing purposes. The first record contains the words 'EFPTR', 'KIND', and 'IDENT'. Each following record contains the Exchange Format identifier followed by its associated kind and ident.

DDNAME : MGTAB

Data Set Name : CAD5.PDDI.V30.TRNDAT

Used By : PRE, POST

Description:

Contains messages output by the Translator. Each message is contained on two records in the file. The first record for each message contains the message identifier, followed by the message type, followed by a length value. The message identifier is a character string of length 5. The message type is a character string of length 1. The length is an integer value of length 4. Valid message types are as follows:

- I - indicates that an integer value is to be printed following the message

- R - indicates that a real value is to be printed following the message

- S - indicates that a string is to be printed following the message

- X - indicates that no associated value is to be printed.

The length value indicates how long the associated value could be if provided. The length value is always 0 if the message type is X.

The second record of each message contains the character message to be printed.

The last record of the file contains only the word 'ENDFL'.

DDNAME : OUTFIL

Data Set Name : &USER_ID.WF&TIME.WFFILE

Used By : Working Form Printout

Description:

The Working Form Printout is output to this file. An example of the Working Form Printout can be found in the PDDI Translator User's Manual, Appendix B.

DDNAME : PASFIL

Data Set Name : &USER_ID.R&TIME.RECORD

Used By : PRE, POST

Description:

Contains user provided information required by the system. This file is formatted as follows:

record number	value	
1	'PRE OR POST'	
2	'PRE' or 'POST'	indicates choice of pre- or post-processing made by the user.
3	'DRAWING NAME'	
4	file/retrieval key	If pre-processing and internal filing, then a valid file/retrieval key is needed. If post-processing and internal filing, then a blank is needed. If using PDDI supplied database software, input "SM00000".
5	'BATCH OR INTERACTIVE'	
6	'B' or 'I'	Indicates choice of batch or interactive processing made by the user.
7	'DISK/TAPE'	
8	'DISK' or 'TAPE'	Indicates choice made by user of device where Exchange Format resides or will reside.

If POST and DISK chosen:

9	dataset	Dataset holding Exchange Format to be processed.
---	---------	---

10	'WF' or 'NO'	Indicates choice of Working Form Printout or no printout.
----	--------------------	---

11	'MAPPING FILE'	
----	----------------	--

12	'Y' or 'N'	Indicates choice of mapping file or no mapping file.
----	------------------	--

If POST and TAPE chosen:

9	tape name	
---	-----------	--

10	tape label	
----	------------	--

11	'WF' or 'NO'	Indicates choice of Working Form Printout or no printout.
----	--------------------	---

12	'MAPPING FILE'	
----	----------------	--

13	'Y' or 'N'	Indicates choice of mapping file or no mapping file.
----	------------------	--

If PRE and DISK chosen:

9	dataset	Dataset holding Exchange Format to be processed.
---	---------	---

10	MAPPING FILE'	
----	---------------	--

11	'Y' or 'N'	Indicates choice of mapping file or no mapping file.
----	------------------	--

If PRE and TAPE chosen:

9	tape name	
---	-----------	--

10	tape label	
----	------------	--

11	MAPPING FILE'	
----	---------------	--

12	'Y' or 'N'	Indicates choice of mapping file or no mapping file.
----	------------------	--

DDNAME : TTYIN

Data Set Name : Set To Default Input

Used By : Working Form Printout

Description:

User input Working Form Printout choices through this DDNAME if processing is done interactively.

DDNAME : TTYOUT

Data Set Name : Set To Default Output

Used By : Working Form Printout

Description:

Outputs prompts for Working Form Printout software.

APPENDIX D

PDDI SYSTEM MESSAGE TABLE

Translator Message Table

Associated with each message is an identifier that indicates where the message was generated.

PDXX	Post-processor
POXX	Post-processor
PRXX	Pre-processor
SYXX	System

The following two message identifiers indicate that the message will contain a value produced by the processor:

VPOXX	Post-processor
VPRXX	Pre-processor

MESSAGE TABLE

<u>Code</u>	<u>Message</u>
PD01	KIND TABLE NOT FILLED, POST-PROCESSOR QUIT
PD02	UNABLE TO GET TOKEN FROM EXCHANGE FILE, POST-PROCESSOR QUIT
PD03	TOKEN NOT PROCESSED:
PD04	WARNING: FAILURE IN HEADER SECTION
PD05	WARNING: FAILURE IN DECLARATION SECTION
PD06	FAILURE IN DATA SECTION, WENT TO NEXT ENTITY
PD07	EXCHANGE FORMAT RECORD:
PD08	CAN'T GET TO NEXT ENTITY, POST-PROCESSOR QUIT
PD09	END OF FILE ENCOUNTERED BEFORE LOGICAL END OF EXCHANGE FORMAT
PD10	NOT A VALID TOKEN:
PD11	WARNING: SEMI-COLON DOES NOT FOLLOW ENDSEC KEYWORD
PD12	SKIPPED TO NEXT ENTITY
PD13	UNABLE TO READ EXCHANGE FORMAT RECORD

SY12 DATABASE RETRIEVE SUCCESSFUL
SY15 BAD CONVERSION - TRANSLATION NOT COMPLETED
SY20 BAD ALLOCATION FROM MDYN - PROGRAM ENDING
SY21 BAD KEY-CANNOT OVERFILE. TRYING TO FILE AGAIN. IDB RETURN
COD
P00 POST-PROCESSOR COMPLETED
P01 NUMBER OF WORKING FORM ENTITIES PROCESSED:
PR0 PRE-PROCESSOR COMPLETED
PR3 PRE-PROCESSOR ENDED DUE TO ERROR
PR7 PRS NOT CONVERTED TO RS
PR8 PRS SUCCESSFULLY CONVERTED TO RS
PR9 FOUND NO PRS TO CONVERT
PR10 UNABLE TO MATCH KIND OF ENTITY NAME. KIND:
PR11 UNABLE TO UPDATE MAPPING OF KIND/IDENT TO EF IDENTIFIER.
PR12 UNABLE TO QUERY MAPPING OF KIND/IDENT TO EF IDENTIFIER.
VP08 TOTAL EXCHANGE FORMAT ENTITIES TRANSLATED:
VP010 FILE RETRIEVAL KEY FOR THE WORKING FORM MODEL IS
VP015 ENTITY NOT TRANSLATED:
VPR4 EXCHANGE FORMAT ENTITY NOT CREATED:
VPR6 ERROR IN WRITING THE RECORD
VPR7 ERROR IN READING THE KIND TABLE
VPR8 ERROR IN READING THE PUNCTUATION

Access Software Message Table

Messages associated with installing the Access Software are computer system unique and are not provided in this manual.